

Realtime Hierarchical Clustering based on Boundary and Surface Statistics

Dominik Alexander Klein¹, Dirk Schulz¹, and Armin Bernd Cremers²

¹ Fraunhofer FKIE, Dept. Cognitive Mobile Systems

² Bonn-Aachen Int. Center for Information Technology (b-it)

May 27, 2016

Visual grouping is a key mechanism in human scene perception. There, it belongs to the subconscious, early processing and is key prerequisite for other high level tasks such as recognition. In this paper, we introduce an efficient, realtime capable algorithm which likewise agglomerates a valuable hierarchical clustering of a scene, while using purely local appearance statistics.

To speed up the processing, first we subdivide the image into meaningful, atomic segments using a fast Watershed transform. Starting from there, our rapid, agglomerative clustering algorithm prunes and maintains the connectivity graph between clusters to contain only such pairs, which directly touch in the image domain and are reciprocal nearest neighbors (RNN) wrt. a distance metric. The core of this approach is our novel cluster distance: it combines boundary and surface statistics both in terms of appearance as well as spatial linkage. This yields state-of-the-art performance, as we demonstrate in conclusive experiments conducted on BSDS500 and Pascal-Context datasets.

1 Introduction

One of the major challenges in computer vision is the question of semantic image partitioning. While today's cameras do record impressive numbers of pixels per image, the amount of possible segmentations and subdivisions of such pictures is even incredibly much higher. Therefore, generation of coherent parts and object candidates is a crucial step in every vision processing pipeline prior to higher level semantic interpretation.

There are several algorithmic variants how to break down the data for semantic classification and object detection: For instance, the family of plain window scanning methods has become popular in classification of certain object types [38, 10, 13]. Usually, a vast

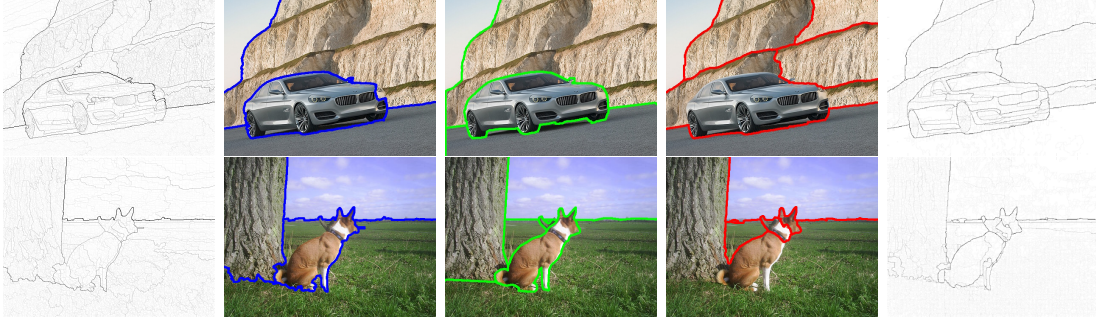


Figure 1: Exemplary results from Pascal-Context dataset [24]. From left to right per row: ultrametric contour map (UCM) and optimal image scale (OIS) segmentation of our approach RaDiG (blue), ground-truth boundaries (green), OIS and UCM of the leading approach MCG-UCM [5](red).

amount of candidates is sampled regularly without respect to the image content itself. Still, such methods have issues generating the proper candidates, e.g. with objects that do not fit well in rectangular shapes. A second group of candidate generating methods is based on interest point operators, which first locate prominent points before aligning known shapes with respect to matched ones [22, 33, 20]. This usually generates less and high quality object candidates. However, it requires an object model to transform the set of localized interest points into object candidates. This is a significant drawback when dealing with a large number of different part- and object classes or priorly unknown ones to be found. Unsupervised clustering is a third line in this taxonomy of approaches and the most biologically inspired one, since it resembles the mechanism of visual grouping, which has been explored by psychologists and neurophysiologists for decades [15, 39]. Such methods can provide a complete partitioning of the scene arising from the data itself. Even more, some approaches yield a hierarchy of nested segmentations, which naturally corresponds to object-part relations. In essence, these methods rely on a proper distance metric to measure differences between image parts. The general hypothesis is that a semantically meaningful entity is of consistent appearance and in contrast to its surroundings in some way. A comprehensive research about what defines this *consistency* in human perception was performed by the Gestalt school of psychology [41] and influenced many technical approaches.

The algorithm introduced in this paper belongs to the unsupervised clustering approaches and is named RaDiG (**R**apid **D**istributions **G**rouping). At its core, it follows the well known agglomerative clustering paradigm, greedily merging the pair of *closest* segments in each step up to total unification, but innovating in many details. Our main contributions comprise

- an innovative, threefold distance metric incorporating boundary contrast with surface dissimilarity as well as spatial linkage,
- the texture-aware enhancement of Ward’s minimum variance criterion with the

meaningful and efficient Wasserstein distance in the space of normal distributions of features,

- a considerable reduction of runtime due to several algorithmic tweaks with pruning and maintaining the cluster connectivity graph as well as an economic feature processing.

In total, these improvements result in an expeditious algorithm which meets strict real-time requirements. Still, the quality of results is on level with the best, globally optimized and highly complex approaches (cf. Fig. 1). Therefore, we claim that it is the method of choice in mobile use cases such as robotics. This is supported by our experimental findings presented in sections 4 and 5.

2 Related Work

There is a huge history of research concerned with segmentation in general settings as well as more specific in computer vision. Here, we restrict our review to the most related, seminal and/or up-to-date methods, and in addition referring the reader to appropriate survey literature. Unsupervised clustering is a field at least as old as computer science itself. We recommend the textbook of Hastie, Tibshirani and Friedman [17] as a comprehensive work about algorithms and statistical background. More specialized on agglomerative hierarchical clustering and thus related to this paper is the overview written by Murtagh and Contreras [25].

Image segmentation in computer vision is a wide subject with many applications. In this paper, we narrow our interest down to methods which segment natural images into semantically meaningful entities in a fully unsupervised fashion, i.e. without any prior knowledge about the image content or objects.

There are "planar" methods, which aim to find an optimal partitioning of the input image. An example is the famous mean shift approach [8], in which pixels are assigned to the next stationary point of an underlying density in the joint domain of color and image position. Some planar methods do integrate a notion of a best segmentation across all scales of structures in the image [9, 14]. Recently, the Superpixel paradigm became popular, which focuses on segmenting the basic building blocks of images [1, 36]. This usually attains an oversegmentation of semantic entities, therefore some approaches add a further accumulation step of Superpixels into larger clusters [19, 43].

Besides planar clustering techniques, there are methods providing a hierarchical tree (dendrogram) of nested segmentations. This additionally determined subset-relation can be useful for further processing. With such hierarchy, the scale of results is not fixed: the nodes close to the root represent a coarse segmentation, which becomes a fine oversegmentation traversing the tree to the leaves [26]. Most approaches can be categorized into either divisive or agglomerative type. Divisive ones build the hierarchy in a top-down manner: iteratively, the current partitioning is split into a finer one. For each step, one could apply a planar segmentation algorithm on each remaining cluster, until they show a uniform appearance. An advantage of divisive methods is that they



Figure 2: RaDiG processing steps. From left to right: input image, gradient magnitude, watershed segments, initial cluster distances, final ultrametric contour map, optimal image scale (OIS) segmentation.

can naturally exploit the statistics of an entire image/region to find an optimal splitting. Among such methods, there is the family of normalized cuts based approaches [9, 32]. In contrast, agglomerative approaches merge initial clusters in a bottom-up way until complete unification. Region based approaches maintain and match feature statistics of a segment’s surface, such as the mean color or a histogram of texture [37, 7, 2]. As Arbeláez [3] pointed out, a hierarchy tree is the dual representation of an ultrametric contour map (UCM), if the underlying metric holds the *strong triangle inequality*. With this insight, the way of thinking image segmentation shifted to concentrate on contour detection. This fruitful ansatz led to a family of algorithms which globalize the results of sophisticated contour detectors using spectral methods, before a greedy agglomerative clustering constructs the hierarchy based on average boundary strengths [4, 5, 34]. Since a planar segmentation is achievable at lower computational complexity than hierarchical approaches, for the sake of processing speed it can be advantageous to first compute an initial oversegmentation, before constructing a hierarchy on top [16, 23, 18]. In a similar way, contour based approaches benefit from an atomic oversegmentation since it can significantly accelerate the globalization of boundary strengths [34]. However, the spectral contour globalization process can be speed up alike by iterated decimation and information propagation of the affinity matrix [5].

Our algorithm RaDiG also provides a hierarchy of segments for further processing and is designed to be realtime capable. Thus, we have chosen the greedy framework of agglomerative clustering starting from a topological Watershed transform. While this basis is similar to several other approaches, ours is considerably different in its components, such as the performant processing of the image graph structure, the efficient representation of feature distributions, and the profitable combination of novel boundary contrast, region dissimilarity as well as spatial linkage within an innovative cluster-distance.

cluster (node)	<i>hierar. connectivity</i> : parent, left-, and right child cluster-IDs <i>planar connectivity</i> : adj.-list of $\begin{pmatrix} \text{neighbor-ID} \\ \text{boundary-ID} \\ \text{cl.-distance} \end{pmatrix}$; near.-neighbor index <i>statistics</i> : area A in number of pixels; color distribution $\left\{ \begin{pmatrix} \mu_L \\ \mu_a \\ \mu_b \end{pmatrix}, \begin{pmatrix} \sigma_L \\ \sigma_a \\ \sigma_b \end{pmatrix} \right\}$
boundary (edge)	<i>hierar. connectivity</i> : parent boundary-ID <i>statistics</i> : boundary length l ; average contrast $\bar{\delta}$

Table 1: Data stored per cluster and boundary element.

3 Rapid Distributions Grouping

The execution of our method comprises four main steps. At first, we convert the color space of the input image to CIE-Lab. In the following sections, we will name the luminance dimension L and the chromatic dimensions with a resp. b . Next, we compute an oversegmentation of the image (Sec. 3.1). This is used to initiate the finest layer of the hierarchical clustering (Sec. 3.2). Finally, our agglomerative approach greedily merges the pair of *closest* segments in each step up to total unification (Sec. 3.3). Intermediate results of the approach are shown in Figure 2. All gained subset relations are kept in a tree structure to be available for further steps of a processing pipeline.

3.1 Atomic Subdivision using Watershed Transform

Since the number of clusters is doubling with each additional level of the cluster hierarchy, it can potentially save a lot of processing time to prune the finest layers from the pixel level and replace them with a planar, atomic subdivision. An established method producing such oversegmentation based on irregularities in the image function is the watershed transform. We apply a variant of the topographical watershed transform based on hill climbing [30] which features $\mathcal{O}(n)$ runtime complexity in the number of pixels and is very fast in practice. It operates on the gradient magnitude of the input image, which we determine as follows: the partial image derivatives in x and y directions are calculated by convolution with the optimized, derivative 5-tap filters of Farid and Simoncelli [12] for each layer L , a , and b . Then, our Lab gradient magnitude is made of independent luminance and chromaticity parts

$$|\nabla| = \sqrt{L_x^2 + L_y^2} + \sqrt{2(a_x^2 + a_y^2 + b_x^2 + b_y^2)}. \quad (1)$$

There is a complete field of research on edge strength calculations by itself. While recently machine learning based methods have shown to improve bounding edge detections [42, 21], that sophisticated and costly quantification is not crucial or often not even beneficial when initializing an oversegmentation.

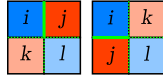


Figure 3: The boundary segment between i and j is a diagonal one, if the labels of i with l or j with k are equal.

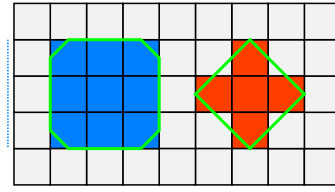


Figure 4: Both blue and red clusters have a boundary length of 12 counting in \mathbf{L}^1 -norm. With shortening, \mathbf{L}^2 -norm is approximated to length 10.82 for the blue and 8.49 for the red cluster.

3.2 Founding of the Hierarchy Tree Structure

The watershed transform uniquely labels each pixels such that it results in a set of connected components each denoting an atomic building block of the image. In order to setup the graph based hierarchical clustering, we need to gather region adjacencies and feature statistics. Our graph structure is made of indexed sets of cluster (node) and boundary (edge) objects, each containing the information listed in table 1. With a single scan through the image and labels, one could assign this information. The hierarchical connectivity is initialized by a constant NOT_CONNECTED. Each pixel joins its corresponding cluster's size and color statistics. If the label to the right or bottom is different from the current, we have to initiate/update the corresponding boundary element:

- Let i and j be the labels and w.l.o.g. $i < j$, we look for the first neighbor-ID $\geq i$ in the (sorted) adjacency list of cluster j . If we found i , we increase the border length of b_{ij} by 1, else we initiate a new boundary of length 1 and add the link ahead of the found position in j 's adjacency list.
- We check the labels below/right for presence of a diagonal boundary (cf. Fig. 3). If found, we shorten the border length of b_{ij} by $\sqrt{2}/2$ in order to approximate \mathbf{L}^2 -norm (cf. Fig. 4).

Then, we iterate the clusters once: for each neighbor in the current cluster's adjacency list, we fill in the value of our cluster-distance and insert the backlink at the end of the neighbor's adjacency list. Note that this operation retains the correct sorting of adjacency lists, because all neighbor-IDs were bigger and the clusters are processed by increasing index. Furthermore, we bookmark the nearest neighbor per cluster with respect to cluster-distances.

The runtime complexity of this tree foundation is $\mathcal{O}(n)$, since we visit each pixel only once and the individual operations are of constant complexity. Please note that the degree of a node with respect to neighborhood edges (its adjacency list size) is constant on average (< 6) as a conclusion from "Euler's planar graph formula", since only segments touching in the image domain become connected by an edge in the graph.

3.3 Agglomerative Clustering

We impose a graph structure of nodes and edges (cf. Tab. 1) on the image. A node represents a cluster (set of connected pixels). There are two kinds of edges between clusters: the first are planar edges, which represent a common boundary between some pair of clusters with respect to 4-connectedness of comprised pixels and when existing at the same *time* in the hierarchical tree. In this sense, a node's lifetime spans from its creation until it is merged. Thus, the timeline is made of the chronological order of merge events in the hierarchy tree. The others are hierarchical edges that connect a pair of merged segments with their parent. These hierarchy edges form a binary tree with the root node representing the whole scene down to the leaves being atomic clusters resulting from the preceding watershed transform.

At each step, agglomerative clustering chooses the cluster pair of lowest distance to be merged. Therefore, all current candidate pairs are maintained in a priority queue ordered by their distance. For this, we employ a Fibonacci heap implementation. Here, the DELETE operation is most costly with $\mathcal{O}(\log m)$ runtime complexity in the number of heap entries. Since every merged pair has to be deleted from the heap once, the overall complexity is dominated by $\mathcal{O}(n \log m)$. Thus, to optimize the runtime, we have to keep the number m of candidates in the heap as low as possible. From section 3.2 we already know that $m < 3n$ even if we consider all planarly connected clusters. Pushing only the nearest-neighbor connected to each cluster to the heap would guarantee $m \leq n$. We can improve this further to $m \ll n/2$ pushing only reciprocal nearest neighbor (RNN) pairs to the heap, since this is a necessary condition for the pair of *globally lowest* distance. Unfortunately, a further improvement which caches nearest-neighbor-chains¹ (NNC) is not applicable, since the local connectivity violates the *reducibility* prerequisite [6].

Each time two clusters are merged, a new parent cluster is initialized with the joined statistics and adjacencies. Merging of two sorted adjacency lists into one is of linear time in the number of entries, thus amortized constant in this case. Furthermore, this way the case where a cluster was neighboring both children can be easily identified and treated in special way: the two boundaries between the cluster and both children are concatenated, in other words a new boundary object with added lengths and weighted harmonic mean of contrasts is created to replace them. The cluster-distances between the parent and its neighbors are calculated, nearest-neighbor edges identified and finally corresponding RNNs updated in the heap. When only a single active cluster is left over, it represents the whole scene and is the root of the finalized hierarchy tree.

3.4 Cluster-Distance from Boundary Contrast and Surface Dissimilarity

While handling of data structures is important for efficiency, the quality of the hierarchical segmentation is determined by the design of a proper cluster-distance. Hence, this is one of our main contributions. Our distance function \mathcal{D} is threefold, comprising the

¹https://en.wikipedia.org/wiki/Nearest-neighbor_chain_algorithm

following parts:

$$\begin{aligned}\mathcal{D}(P, Q) &= \log(\omega(P, Q)) && (\text{surface dissimilarity}) \\ &+ \log(\bar{\delta}(P, Q)) && (\text{boundary contrast}) \\ &+ \log(\eta(P, Q)) && (\text{spatial linkage}).\end{aligned}\tag{2}$$

Here, P and Q denote clusters with a common boundary. All three parts are innovative in some aspect. In the following subsections, we will explain each term in more detail. Please note that adding the logarithms is a geometric fusion of terms (equivalent to a product), hence one does not need to normalize the scales of individual parts to a common range. Nonetheless, it is suggestive to train weights for an optimized combination of parts. This could further improve results at virtually no costs, but was not yet exploited for experiments in this paper.

Surface Dissimilarity Term Bucking the trend, our approach is very thrifty in computing different kinds of features. Indeed, we only use a normal distribution of colors to describe the appearance of a cluster (cf. Tab. 1). These statistics are gathered in the form of ML-estimates from the individual colors of all pixels belonging to a certain cluster. We primarily came to this decision, because the joining of Gaussian statistics is a very efficient, constant time operation [35]. The second good reason is that the Wasserstein distance between normal distributions is meaningful and fast to compute. The Wasserstein distance is a transport metric between probability distributions. It accounts how much (probability-) mass needs to be carried how far wrt. an underlying metric space. Here, the underlying space is CIE-Lab with \mathbf{L}^1 -norm², which mimics human texture discrimination in a simplified way when working on top of the perceptually normalized CIE-Lab colorspace [31]. Using normal distributions, the 1st Wasserstein distance solves to the expression

$$\mathcal{W}_1(\mathcal{N}_P, \mathcal{N}_Q) = |\vec{\mu}_P - \vec{\mu}_Q| + \left| \text{tr}(\sqrt{\Sigma_P}) - \text{tr}(\sqrt{\Sigma_Q}) \right|. \tag{3}$$

Ward's clustering criterion [40] is a distance which estimates the growth in data variance when joining clusters,

$$\text{Ward}(P, Q) = \frac{d(\mu_P, \mu_Q)^2}{1/A_P + 1/A_Q} \tag{4}$$

where A refers to the clusters' surface areas in pixels. If you think of this in terms of color means, the intuition is that mixing more and more colors always ends up in some grayish tone, but then subtle differences between such meshes can still make a clear difference for large surfaces. This measure is known for developing clusters of more balanced sizes. As a novelty, we propose to lift this concept from color to texture by replacing the distance between means with the 1st Wasserstein distance, which yields

$$\omega(P, Q) = \frac{\mathcal{W}_1(\mathcal{N}_P, \mathcal{N}_Q)^2}{1/A_P + 1/A_Q}. \tag{5}$$

²This is similar to the well known earth mover's distance (EMD) on histograms, which is in fact the discretized \mathcal{W}_1 distance.

Boundary Contrast Term Each boundary element carries information about its average (weighted harmonic mean) boundary contrast $\bar{\delta}(P, Q)$ between the clusters on either side. This value is kept up-to-date if two boundary elements are concatenated during a merge event (cf. Sec. 3.3). The remaining question is how it should be initialized. Instead of using the gradient magnitude or related measures on the contour at pixel resolution, we decided to plug in the Wasserstein distance between P and Q . Obviously, this is most economic since it is already computed as a part of the surface dissimilarity. But there are more good reasons for this decision: a measure incorporating the area alongside the contour-line better assesses the strength of blurry edges often emerging from motion or defocussing. Furthermore, we smartly avoid cross-talk artifacts at the crossings of edges, a problem Arbeláez et. al. [4, 5] explicitly addressed by a procedure called Oriented Watershed Transform (OWT). Please note that despite re-using the Wasserstein distance, the boundary contrast is essentially different to the surface dissimilarity: from the way boundaries and clusters are merged, the former is the average difference between the opposing, atomic clusters along a boundary, while the latter includes the difference between both entire surfaces.

Spatial Linkage Term From graph topology, it is a binary decision if two segments are connected by an edge or not. However, it is reasonable to consider a certain factor of how close they actually are. We introduce the simple yet effective connectivity term

$$\eta(P, Q) = \left(\frac{\sqrt{A_P}}{l_{PQ}} \cdot \frac{\sqrt{A_Q}}{l_{PQ}} \right)^{1/2} = \frac{(A_P \cdot A_Q)^{1/4}}{l_{PQ}}, \quad (6)$$

which combines the extents of the common boundary and both surfaces. This way, the linkage between clusters is rated independently of concrete positions and thus is very flexible in shape. Relating the radical of the cluster surface to the common boundary's length is a normalized measure of connectivity, but advances versus simply using the perimeter by preferring more convex, smooth shapes versus elongated or jagged ones. We put the geometric mean of both surfaces' score, since it devaluates a cluster pair more towards the inferior result and thus better avoids residues staying alive.

4 Evaluation of Key Contributions

Since we introduced several contributions in this paper, it is reasonable to evaluate how much the overall result depends on every single innovation. Therefore, we compare different variants of our system each with one enhancement deactivated or replaced. The experiments are conducted on the trainval images of the BSDS500 benchmark [4]. This dataset provides boundary annotations of several human subjects, which were interpreted to define a segmentation.

Segmentation Quality Measure It is not straightforward to rate the quality of a segmentation with respect to a ground truth. Pont-Tuset and Marques [27] defined the F_{op} measure in the context of generating object or part candidates for recognition. Basically, this allows a segment to be fragmented into several parts. Regions need to exceed some relative overlaps to be accepted as valid object resp. part matches. Those matches are

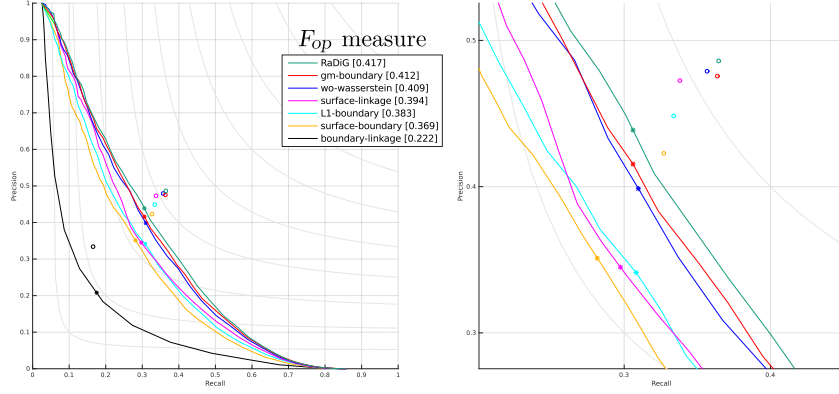


Figure 5: Precision/recall curve of RaDiG and its reduced variants on BSDS500 trainval set using the object-and-parts F_{op} measure. The optimal dataset scale (ODS) is drawn with an asterisk on each curve, while the optimal per image scale (OIS) is drawn with a circle and numbers are given in the legend.

weighted by their fraction of overlap, thus, parts are only partially counted as true positive matches. On this basis, a precision and recall for object-and-part matches is defined. Varying a threshold on the cluster-distance of the cluster hierarchy (or UCM) produces different levels of segmentations from coarse to fine. Coarse segmentations are likely of low precision, but high recall, and vice versa comparing fine oversegmentations. The F_{op} measure is the maximally achievable value of the harmonic mean of these precision and recall pairs.

Boundary Contrast Evaluation The RaDiG boundary contrast is initialized with the Wasserstein distance between neighboring, atomic clusters and further on averaged with respect to approximated L^2 -norm length when two parts are concatenated. To assess if our novelties really contribute to the overall results, we conducted two experiments: Figure 5 shows the original algorithm’s result (RaDiG) in comparison to a variant where we averaged the gradient magnitude along the boundary in order to initialize the boundary contrasts (gm-boundary), which is more expensive to compute but slightly inferior. We also tried L^1 -norm lengths (Fig. 5, L1-boundary) for averaging the boundary contrast, but here the little additional effort for approximation pays off well.

Appearance Distance Evaluation Next, we assess the benefit of integrating covariance information using the Wasserstein distance. Thus, we replace it with usual squared Euclidean metric between the means, yielding the classic Ward clustering criterion and a simpler boundary contrast (Fig. 5, wo-wasserstein). We deduce that the Wasserstein distance is a worthwhile enhancement, raising the F_{op} from 0.409 to 0.417 and showing a consistently better trend on the precision/recall curves. From our observation, the difference is pronounced with larger segments, when textures are more often of non-uniform color.

Threefold Cluster-Distance Evaluation Our agglomerative clustering approach efficiently maintains statistics from both the boundaries between and the surfaces of clus-

resolution	#pixel	time in ms
135×90	12150	11
270×180	48600	41
320×240	76800	64
481×321	154401	107
540×360	194400	151
1080×720	777600	604

Table 2: Runtime of RaDiG on a single core of an Intel Xeon X5680 @ 3.33GHz for different resolutions.

component	time in ms	time in %
colorspace	10	16
watershed	7	10
founding tree	20	32
agg. clustering	27	42
\sum serial	64	100
\sum parallel	27	42

Table 3: Runtime of different components of our RaDiG approach at QVGA resolution (320×240).

ters, which gives rise to our sophisticated, threefold cluster-distance term (cf. Eq. (3)). In this experiment, we figure out if all three components contribute to the fine overall results, by omitting one of the parts at a time. We named the twofold cluster-distances by the remaining components: surface-linkage, surface-boundary, and boundary-linkage (cf. Fig. 5). Dropping one of the parts severely worsens the results. Particularly important is the Ward-based term of surface dissimilarity, probably because it controls the clusters’ sizes not to diverge too much during agglomeration.

Runtime Evaluation We claimed that our implementation is realtime capable and therefore the best choice for applications on mobile platforms such as autonomous robots. Table 2 shows the overall runtime of our approach for different resolutions, estimated as an average from six cluttered, natural images. Empirically, it seems as if, thanks to pruning of candidate pairs in the heap data structure (cf. Sec. 3.3), we achieved an amortized linear behavior in the number of pixels. For QVGA resolution, table 3 breaks down how much time each component of our approach consumes. Since our implementation is based on the ROS-framework [29] and every component is implemented as a processing node using its own thread, only the slowest component restricts the framerate when processing video data. Thus we are able handle up to 37 fps in QVGA on our machine. On BSDS500 image size (481×321), the single thread variant of RaDiG takes on average 107ms to run the whole algorithm. Thereby, it outperforms the approaches of Arbeláez et. al. [5] by one resp. two orders of magnitudes. Testing their code of version 2.0 on the same machine, we measured runtimes of 17.6 seconds for MCG-UCM (165× slower) and 2.8 seconds for SCG-UCM (26× slower) on BSDS500 data.

5 Comparative Experiments on BSDS500 and Pascal-Context

We compared absolute performances and ranked our approach versus state-of-the-art ones on widely used datasets and appropriate measures. Pont-Tuset and Marques [27] provide a number of results from different algorithms based on their F_{op} measure. In addition, we wrote a tool to convert data from the result formats of Arbeláez et. al. [5] plus Ren and Shakhnarovich [28] to update their comparison with recent approaches. We are convinced that F_{op} is the most adequate performance metric for our kind of application, since we aim to perceive semantically meaningful entities. Our method is not an edge detector in the first place, nevertheless we include the F_b measure, which

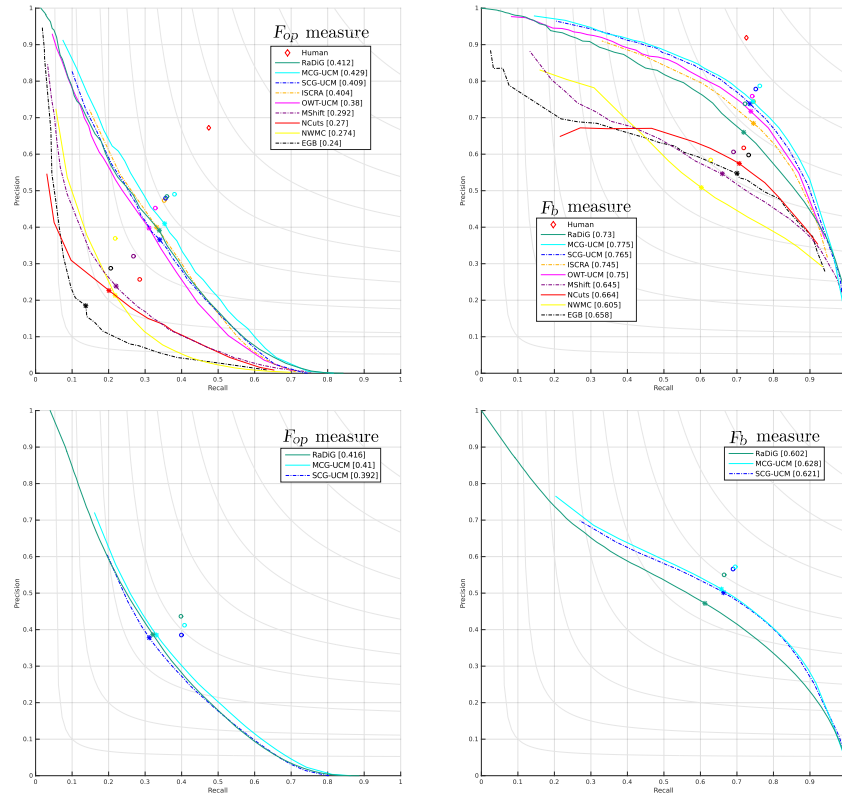


Figure 6: Precision/recall curves of RaDiG and leading approaches using F_{op} as well as F_b measure. Top row: results on BSDS500 test images; second row: results on Pascal-Context dataset. The optimal per image scale (OIS) values (circles) are given in the legend.

quantifies how well contours are reproduced, as additional assessment. Figure 6 first row shows the precision/recall curves for F_{op} and F_b measures of all tested approaches on the test images of BSDS500. There is a noticeable gap between EGB (efficient graph based method of Felzenszwalb and Huttenlocher [14]), NCUTs (normalized cuts [9]), MShift (mean shift [8]), and NWMC (binary partition tree [37]) on one hand, but the recent approaches OWT-UCM (global probability of boundary with oriented watershed transform followed by agglomerative merging [4]), ISRA (image segmentation by cascaded region agglomeration [28]), SCG-UCM (single-scale combinatorial grouping [5]) MCG-UCM (multi-scale combinatorial grouping [5]) and our proposed RaDiG method on the other hand. A clear difference between OWT-UCM, ISRA, SCG-UCM and MCG-UCM versus the others is that they tuned certain aspects of their algorithms using machine learning techniques. Also, they are rather costly to compute and do not target realtime applications. This said, results of RaDiG are very encouraging and our algorithm is presumably the best choice for realtime tasks in unconstrained environments, such as

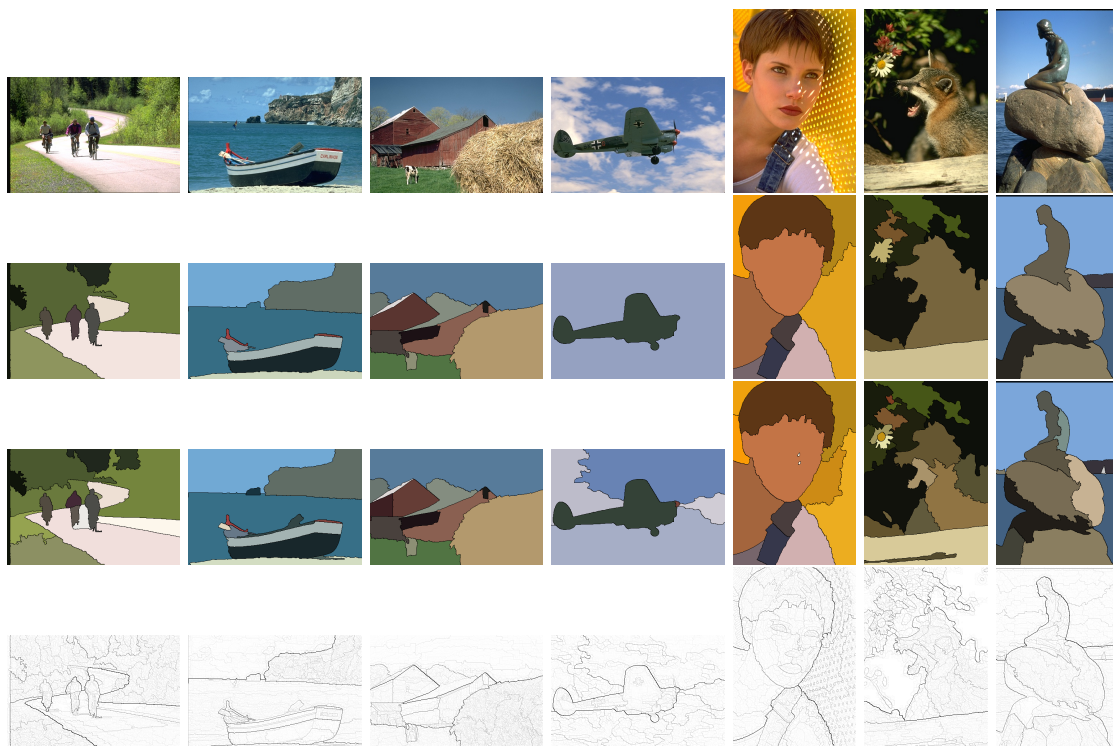


Figure 7: Exemplary segmentations of RaDiG on BSDS500 images. From top to bottom per column: original, optimal image scale (OIS), optimal dataset scale (ODS), and ultrametric contour map (UCM) of the corresponding hierarchy.

often present in autonomous robotics. Figure 7 shows some characteristic outcome of RaDiG on BSDS500 images.

While the BSDS500 dataset is the popularly accepted benchmark in this kind of unsupervised image segmentation, we felt that the ground truth is not entirely appropriate for evaluation of object and part candidates. The Pascal-Context dataset [24] contains pixel-precise labels for more than 400 object categories on the train/val Pascal VOC 2010 data [11] (10103 images), which is appealing also for comparison of unsupervised approaches due to this variety and size. We wrote a tool to convert this ground truth labels into segmentations by separating labels of the same category into spatially connected components, numbering them increasingly and convert them into the ground truth format of BSDS500. Fortunately, two of the best performing approaches on BSDS500, SCG-UCM and MCG-UCM, published their ultrametric contour maps online³ for the main Pascal VOC 2012 images, which is a superset of 2010, so that we are able to compare results. All three approaches retain their very good performance on this large dataset (cf. Fig. 6, second row). However, RaDiG supersedes SCG-UCM on the precision/recall curve and the gap between MCG-UCM and RaDiG is even smaller than

³<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/mcg/>

method	BSDS500		Pascal-C	
	F_{op}	F_b	F_{op}	F_b
RaDiG	0.363	0.688	0.351	0.533
MCG-UCM	0.379	0.744	0.356	0.575
SCG-UCM	0.352	0.737	0.342	0.571
ISCRA	0.363	0.714	-	-
OWT-UCM	0.349	0.727	-	-
MShift	0.229	0.598	-	-
NCuts	0.213	0.634	-	-
NWMC	0.215	0.552	-	-
IIDKL	0.186	0.575	-	-

Table 4: Optimal dataset scale(ODS) results for both comparative experiments.

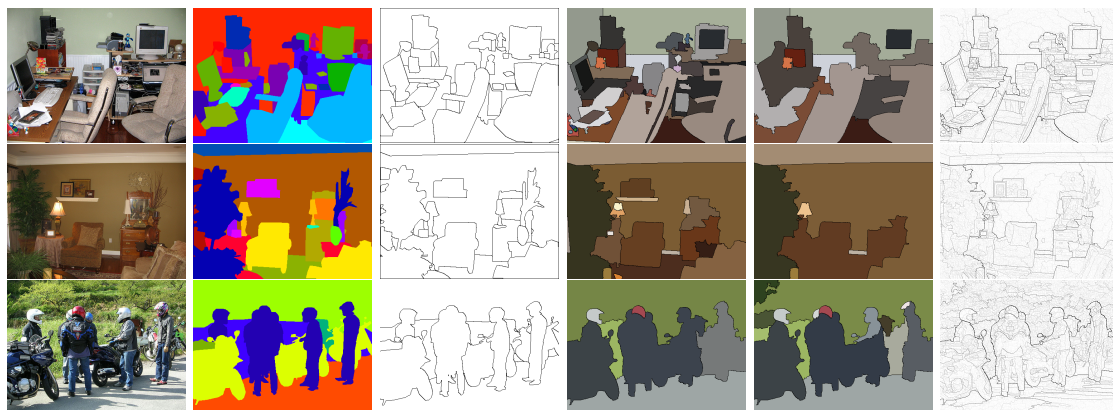


Figure 8: Exemplary segmentations of RaDiG on Pascal-Context images. From left to right per row: original, ground truth labels, g.t. segmentation, optimal image scale (OIS), optimal dataset scale (ODS), and ultrametric contour map (UCM).

on BSDS500. Furthermore, looking at F_{op} from optimal image scale, RaDiG took the lead. However, RaDiG is only moderately suited for precise contour detection, as F_b results suggest. Table 4 reports numbers for the optimal dataset scale results of all approaches, which are also marked in the Figures 5 and 6 by asterisks. Figure 8 shows some qualitative results of RaDiG on Pascal-Context dataset.

6 Conclusion and Future Work

We proposed RaDiG (Rapid Distributions Grouping), a fast and greedy algorithm to construct a tree of nested image segmentations starting from an atomic subdivision. Whereas its overall clustering framework is approved in awhile, we contributed several valuable findings. The feature computations and handling of the graph structure were optimized to enable a very low runtime needed for realtime applications. At the core of the approach, a novel, threefold cluster-distance was introduced and stepwise shown to advance the quality of clustering. Overall, our method plays in the same league

wrt. precision/recall as current state-of-the-art approaches, but is at least an order of magnitude faster. By now, RaDiG is a parameter free approach. We are confident that involving machine learning techniques to a weighted balancing of components could further improve results. At the moment, there are several projects under development, where we deploy RaDiG on mobile robots, e.g. in the area of object manipulation.

References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012. [3](#)
- [2] S. Alpert, M. Galun, A. Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):315–327, 2012. [4](#)
- [3] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 182–182, June 2006. [4](#)
- [4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011. [4](#), [9](#), [12](#)
- [5] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 328–335. IEEE, 2014. [2](#), [4](#), [9](#), [11](#), [12](#)
- [6] M. Bruynooghe. Methodes nouvelles en classification automatique de donnees taxinomiques nombreuses. *Statistique et Analyse des Donnes*, 3:24–42, 1977. [7](#)
- [7] F. Calderero and F. Marques. Region merging techniques using information theory statistical measures. *Image Processing, IEEE Transactions on*, 19(6):1567–1586, 2010. [4](#)
- [8] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002. [3](#), [12](#)
- [9] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1124–1131 vol. 2, June 2005. [3](#), [4](#), [12](#)
- [10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, June 2005. [1](#)
- [11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.

- <http://host.robots.ox.ac.uk/pascal/VOC/voc2010/workshop/index.html>, 2010. **13**
- [12] H. Farid and E. P. Simoncelli. Differentiation of discrete multidimensional signals. *Image Processing, IEEE Transactions on*, 13(4):496–508, 2004. **5**
 - [13] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Trans. on. Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept 2010. **1**
 - [14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. of Computer Vision*, 59(2):167–181, Sep 2004. **3, 12**
 - [15] E. B. Goldstein. *Sensation and Perception*, chapter Perceiving Objects and Scenes. Cengage Learning, 2009. **2**
 - [16] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *Image Processing, IEEE Transactions on*, 7(12):1684–1699, 1998. **4**
 - [17] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics, 2009. **3**
 - [18] V. Jain, S. C. Turaga, K. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung. Learning to agglomerate superpixel hierarchies. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 648–656. Curran Associates, Inc., 2011. **4**
 - [19] P. Kovesi. Image segmentation using slic superpixels and dbscan clustering. <http://www.peterkovesi.com/projects/segmentation>, Apr 2013. **3**
 - [20] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on statistical learning in computer vision, ECCV*, page 7, 2004. **2**
 - [21] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár. Unsupervised learning of edges. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. **5**
 - [22] D. G. Lowe. Object recognition from local scale-invariant features. In *Int. Conf. on Computer Vision (ICCV)*, pages 1150–1157, 1999. **2**
 - [23] B. Marcotegui and S. Beucher. Fast implementation of waterfall based on graphs. In C. Ronse, L. Najman, and E. Decencière, editors, *Int. Symp. on Mathematical Morphology*, pages 177–186. Springer Netherlands, April 2005. **4**
 - [24] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, et al. The role of context for object detection and semantic segmentation in the wild.

- In *Computer Vision and Pattern Recognition (CVPR)*, 2014 IEEE Conference on, pages 891–898. IEEE, 2014. 2, 13
- [25] F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. *Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012. 3
- [26] B. Peng, L. Zhang, and D. Zhang. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3):1020 – 1038, 2013. 3
- [27] J. Pont-Tuset and F. Marques. Measures and meta-measures for the supervised evaluation of image segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2131–2138. IEEE, 2013. 9, 11
- [28] Z. Ren and G. Shakhnarovich. Image segmentation by cascaded region agglomeration. In *Computer Vision and Pattern Recognition (CVPR)*, 2013 IEEE Conference on, pages 2011–2018. IEEE, 2013. 11, 12
- [29] Robotics Foundation. ROS - Robot Operating System. <http://www.ros.org>, 2016. 11
- [30] J. B. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2001. 5
- [31] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000. 8
- [32] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, Aug. 2000. 4
- [33] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Int. Conf. on Computer Vision (ICCV)*, pages 1470–1477, 2003. 2
- [34] C. J. Taylor. Towards fast and accurate segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2013 IEEE Conference on, pages 1916–1922, June 2013. 4
- [35] G. H. G. Tony F. Chan and R. J. LeVeque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247, Aug 1983. 8
- [36] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Computer Vision–ECCV 2012*, pages 13–26. Springer, 2012. 3
- [37] V. Vilaplana, F. Marques, and P. Salembier. Binary partition trees for object detection. *Image Processing, IEEE Transactions on*, 17(11):2201–2216, 2008. 4, 12

- [38] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. [1](#)
- [39] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. A century of gestalt psychology in visual perception i. perceptual grouping and figure-ground organization. *Psychol. Bull.*, 138(6):1172–1217, Nov 2012. [2](#)
- [40] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963. [8](#)
- [41] M. Wertheimer, L. Spillmann, and M. Wertheimer. *On Perceived Motion and Figural Organization*. MIT Press, 2012. [2](#)
- [42] S. Xie and Z. Tu. Holistically-nested edge detection. In *Int. Conf. on Computer Vision (ICCV)*, pages 1395–1403, 2015. [5](#)
- [43] B. Zhou. Image segmentation using slic superpixels and affinity propagation clustering. *Int. J. of Science and Research*, 4(4), Apr 2015. [3](#)